DUW

May 15, 2022

Foundations of Programming, Python

Assignment 05

GitHub: <https://github.com/danielauw/IntroToProg-Python>

**Working with Dictionaries**

# Introduction

In this assignment I will explain step-by-step how I added code to an existing Python program to manage a to do list, allow the user to see the existing data, add tasks based on user input, remove data based on user’s request, and then store the data in a text file called, ToDoList.txt. Assignment 05 is meant to prove understanding of the dictionaries and lists. The last summary section includes my observations and key takeaways from applying Python module 5.

# Creating the Program

The steps I followed to develop and run the program:

1. **Step 1: Opened PyCharm**
   1. Opened the Start menu.
   2. Search for PyCharm and launched the shell.
2. **Step 2: Created a new script file and saved it with the .py extension for Python**
   1. Created a new sub-folder called Assignment 05 inside of the \_PythonClass folder.
   2. Create a new project in PyCharm that uses the \_PythonClass\Assignment05 folder as its location.
   3. Created a python script file within my project called Assignment05.py and saved with the Python file extension.
3. **Step 3: Updated the script header including inline comments in the script file** 
   1. Used the sign symbol # to indicate in Python that it is a comment outside of a code and is not being executed by the program.
   2. I added comments at the beginning of the program to state the title, description of the assignment and change log (who, when, what).
4. **Step 4: Added the code in the script file**

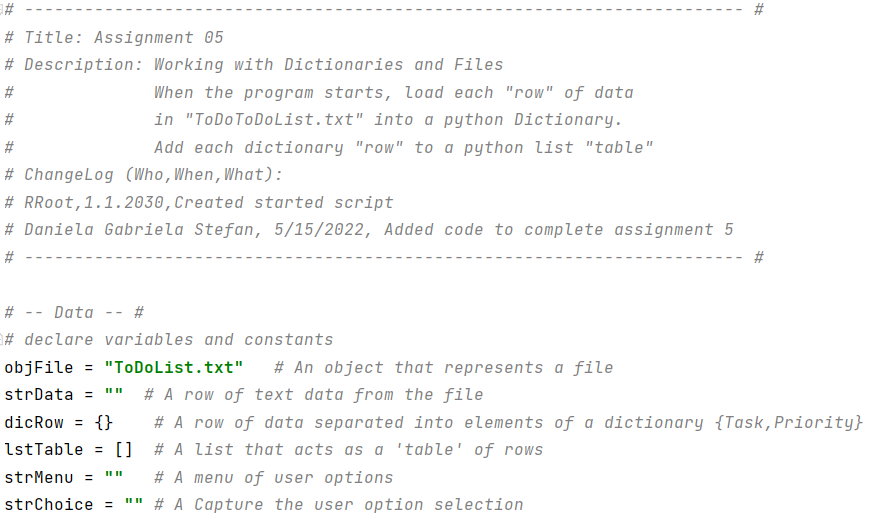
Added the following statements as instructions for Python to execute, each separated by inline comments indicating the steps in the process:

Step 1: Declared variables

In this case, the variables were already declared from the existing code.

***Figure 1*** shows my script so far.

***Figure 1: Script header and declared variables in PyCharm***



Step 2: Loaded the columns into Python Dictionary objects.

Opened the ToDoList.txt file to and gave it the command to read the file.

Defined my columns and rows of data formatted as dictionary objects.

Appended the dictionary objects/rows to the list/table of data.

Step 3: Displaying a menu of choices to the user

In this case, the menu of choices was already in the existing code. The while loop is meant to repeat the print statement (menu) until the user tells it to break (the last choice).

Added an input statement asking the user to make a choice based on the menu.

Added a printed statement to print the option number.

***Figure 2*** shows the last two steps explained above.

Graphical user interface, text, application, email

Description automatically generated

***Figure 2: Defined columns & Menu of choices in PyCharm***

Step 4: Displaying the data in the list each time the user makes that choice

Opened a conditional statement with the first option for the user, i.e., showing current data followed by the following tabbed statements:

A for loop asking the program to go through each row in the table and display each element. In order to avoid displaying the default format of a list with the square brackets, I unpacked the list into individual variables by indicating the position/index subscripts of the variable.

A print statement to display the stored data.

Step 5: Getting user input and adding a new task to the list each time the user makes that choice

Continued the conditional statement to build the second option in the menu, i.e. adding a new task and priority.

Added two input statements to collect information from the user

Appended the user input to the existing list.

Told the program to continue running (the user can stop the program when they select the last option in the menu).

***Figure 3*** shows my code for the first two options in the menu.

Text

Description automatically generated

***Figure 3: First two options in the menu in PyCharm***

Step 6: Getting user input to remove a task from the list.

Continued the conditional statement to build the third option in the menu.

Added an input statements to ask the user what task they would like removed from the list.

Added a for loop that runs through column “Task” in the table to identify the task the user wants removed.

Added code to remove the task.

Added a print statement to indicate to the user that the task has been removed.

Told the program to continue running.

Step 7: Saving the data to a text file when the user makes that choice

Continued the conditional statement to build the fourth option in the menu.

Requested that the ToDoList.txt file open and start writing to the file.

Added a for loop that runs through columns “Task” and “Priority’.

Closed writing to the text file.

Added a print statement for the user.

Step 8: Exiting the program

Continued the conditional statement to build the fifth option in the menu.

Printed a final statement for the user

Told the program to ‘break’ the loop.

***Figure 4*** shows the three above steps.

Graphical user interface, text, application

Description automatically generated

***Figure 4: Options #3 and #4 in the menu in PyCharm***

1. **Step 6: Ran the program through CMD**
2. Opened CMD and added the path file for “Assignment05”
3. ***Figure 5*** shows the output in CMD and ***Figure 6*** shows the output in PyCharm:

Text

Description automatically generated

***Figure 5: Output in the CMD***

Text

Description automatically generated with medium confidence

***Figure 6: Output in PyCharm***

1. **Step 7: Opened the file in a text editor**
2. Located the text file and opened it to verify that it displays the user input
3. ***Figure 7*** shows the output in the text editor

Graphical user interface, text, application

Description automatically generated

***Figure 7: Output in text editor – Verifying that the file has data***

# Summary

Creating and running Assignment 05 allowed me to work with dictionaries and lists. As more concepts are being introduced specifically with regards to data collections, I feel that I need more practice to better understand what code and syntax needs to be used when.